

Modified Frame Reconstruction Algorithm for Compressive Sensing

Graeme Pope

June 5, 2009

Abstract

Compressive sensing is a technique to sample signals well below the Nyquist rate using linear measurement operators. In this paper we present an algorithm for signal reconstruction given such a set of measurements. This algorithm generalises and extends previous iterative hard thresholding algorithms and we give sufficient conditions for successful reconstruction of the original data signal. In addition we show that by underestimating the sparsity of the data signal we can increase the success rate of the algorithm.

We also present a number of modifications to this algorithm: the incorporation of a least squares step, polynomial acceleration and an adaptive method for choosing the step-length. These modified algorithms converge to the correct solution under similar conditions to the original un-modified algorithm. Empirical evidence show that these modifications dramatically increase both the success rate and the rate of convergence, and can outperform other algorithms previously used for signal reconstruction in compressive sensing.

1 Introduction

Compressive sensing is a radical new way of sampling signals at a sub-Nyquist rate. The Shannon/Nyquist sampling theorem tells us that an analogue signal can be reconstructed perfectly from its samples, if it was sampled at a rate at least twice the highest frequency present in the signal, known as the Nyquist rate [28,29]. For many signals, such as audio or images, the Nyquist rate can be very high. This may result in acquiring a very large number of samples, which must be compressed in order to store or transmit them, as well as placing a high requirement on the equipment needed to sample the signal. *Compressive Sensing* (or compressed sensing or CS) is a recently introduced method that can reduce the number of measurements required, in some ways it can be regarded as automatically compressing the signal. Compressive sensing is a technique that enables us to fully reconstruct particular classes of signals if the original signal, is sampled at a rate well *below* the Nyquist rate.

In particular, compressive sensing works with sparse signals. In many applications the signal of interest is primarily zero in some known fixed basis, that is, in this representation the data contained within the signal is sparse. Traditional measurement techniques which consist of sampling at all the possible data points heavily over-sample the signal. Consider the scenario where we randomly draw samples from a sparse signal, then the probability of sampling at an “interesting” data point is equal to the sparsity fraction. Compressive sensing gets around this by using a small number of linear sampling operators that samples the signal across all data points simultaneously. This gives rise to the name compressive sensing as it is a combination of sampling and compression.

More formally, let $\mathbf{x} \in \mathbb{R}^n$ be an s -sparse vector in the basis Ψ , then we can write $\mathbf{x} = \Psi \mathbf{s}$ for some vector $\mathbf{s} \in \mathbb{R}^n$ such that only $s \leq n$ components of \mathbf{s} are non-zero. Assume then that our m linear sampling operators are given by the rows of the matrix $\Phi \in \mathbb{R}^{m \times n}$. The problem of compressive sensing is then to find \mathbf{x} , or equivalently, \mathbf{s} given the measurements $\mathbf{y} = \Phi \mathbf{x}$ and the matrices Φ and Ψ , under the assumption that \mathbf{s} is sparse. If $m < n$ then the problem is under-determined and there is no unique solution, so we commonly say that we are interested in the most sparse solution to this equation, i.e. the minimiser to

$$\min_{\hat{\mathbf{s}} \in \mathbb{R}^n} \|\hat{\mathbf{s}}\|_0 \quad \text{subject to} \quad \mathbf{y} = \Phi \Psi \hat{\mathbf{s}}.$$

To solve this problem in general is known to be NP-hard [17, 26].

One of the original breakthroughs in compressive sensing was to show that linear programming methods can be used to efficiently reconstruct the data signal with high accuracy [11, 12, 19]. Since then many alternative methods have been proposed as a faster or more successful alternative to these linear programming algorithms. One approach is to use matching pursuit techniques, originally proposed in [25], variations have been proposed such as OMP or orthogonal matching pursuit [30], Stagewise orthogonal matching pursuit (StOMP) [18], Compressive sampling matching pursuit (CoSaMP) [27] and gradient pursuit algorithms [5, 4, 7, 8]. Also proposed has been a suite of thresholding based algorithms, either hard thresholding (IHT) [6, 9] or soft thresholding [15, 16]. What we propose is a combination of some of these techniques, combining hard thresholding with matching pursuit methods. We will also show how we can use polynomial acceleration techniques to increase the rate of convergence.

In addition to this, work has also been done on model based compressive sensing in [2], which can be applied to many of the algorithms above. Most of the aforementioned algorithms, in particular CoSaMP and IHT make use of a pruning step which takes a solution and forces it to be sparse, by removing all but the \hat{s} -largest (in magnitude) components, which is the best \hat{s} sparse approximation under any ℓ_p norm for $1 \leq p < \infty$. Model based compressive sensing proposes using a model based algorithm to perform this, that is to choose the sparse signal that is not necessarily closest under an ℓ_p distance, but that best fits the signal model. Such a modification would also be applicable to our algorithm.

We will refer to our algorithm as the ‘‘Modified Frame Reconstruction’’ or MFR algorithm.

1.1 Paper Overview

In section 2 of this paper we will give an overview of the fundamental concepts in compressive sensing. We will also briefly discuss frames, the frame reconstruction algorithm and polynomial acceleration using Chebyshev polynomials. In section 3 we will present our own algorithm and theoretical results regarding its performance. This work is similar to that in [6, 9], but we analyse the algorithm in a different manner and in a more general setting. Later in section 4 we present a number of extensions to this algorithm that increase both the rate of convergence and the probability of successfully estimating the original input signal. We show that by incorporating a least squares step, polynomial acceleration and a variable step-length, that we can significantly increase the performance of our algorithm. In section 5 we perform simulations of our algorithm demonstrating that the theoretical results translate into a real world performance advantage. Section 6 contains a comparison of our algorithm to various other existing methods.

1.2 Notation

In this paper bold-faced symbols represent vectors, such as $\mathbf{x} \in \mathbb{R}^n$, with components x_1, \dots, x_n . For an index set $\Gamma \subset \{1, 2, \dots, n\}$ we take $\mathbf{x}_\Gamma \in \mathbb{R}^n$ to be the vector that agrees with \mathbf{x} on

all the indices $i \in \Gamma$ and is 0 elsewhere. For a matrix $\Phi \in \mathbb{R}^{m \times n}$ we take Φ^\top, Φ^* and Φ^\dagger to be the transpose, complex-conjugate transpose and Moore-Penrose pseudo-inverse respectively. Hence $\Phi^\dagger \mathbf{y}$ is the solution to the minimisation problem $\arg \min_{\mathbf{x}} \|\Phi \mathbf{x} - \mathbf{y}\|_2$. For a matrix Φ we take Φ_Γ to be the submatrix formed by taking the columns of Φ indicated by the set Γ . When we write Φ_Γ^\top we apply the column selection first, then the transpose, e.g. $\Phi_\Gamma^\top \triangleq (\Phi_\Gamma)^\top$. We take $\lambda_{\min}(\Phi), \lambda_{\max}(\Phi)$ to be the lower and upper eigenvalues of the matrix Φ and similarly $\sigma_{\max}(\Phi), \sigma_{\min}(\Phi)$ are the upper and lower singular values of Φ .

Recall that the ℓ_p distance, for $1 \leq p < \infty$, between two vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ is defined to be $d(\mathbf{x}, \mathbf{y}) \triangleq (\sum_{i=1}^n |x_i - y_i|^p)^{1/p}$ giving rise to the ℓ_p norm $\|\mathbf{x}\|_p \triangleq (\sum_{i=1}^n |x_i|^p)^{1/p}$. The ℓ_0 pseudo-norm is given by $\|\mathbf{x}\|_0 \triangleq |\{x_i \neq 0\}|$, i.e. the number of non-zero components of \mathbf{x} . We say that a vector \mathbf{x} is s -sparse if $\|\mathbf{x}\|_0 = s$, that is, precisely s components of \mathbf{x} are non-zero. The i -th stage of an iterative algorithm producing the vector \mathbf{x} is denoted by $\mathbf{x}^{(i)}$.

We will write $\mathbb{H}_\tau : \mathbb{R}^n \rightarrow \mathbb{R}^n$ to be the function that returns $\hat{\mathbf{x}}$, a best τ -sparse vector approximation to the input \mathbf{x} under any ℓ_p -norm, $1 \leq p < \infty$, i.e. a τ -sparse vector such that $\|\mathbf{x} - \hat{\mathbf{x}}\|_p$ is minimal. Note that this has the same solution for any $1 \leq p < \infty$: the vector $\hat{\mathbf{x}}$ consisting of the τ -largest (in magnitude) components of \mathbf{x} . If this vector is not unique, we can decide whether to break the tie randomly or deterministically. In our implementation we chose to break the tie lexicographically. We also write $\text{supp}(\mathbf{x})$ to denote the support of the vector \mathbf{x} , that is, $\text{supp}(\mathbf{x}) \triangleq \{x_i : x_i \neq 0\}$.

2 Background

2.1 Compressive Sensing

Let $\mathbf{x} \in \mathbb{R}^n$ be a signal and let $\Phi \in \mathbb{R}^{m \times n}$ be a measurement matrix. We call the vector $\mathbf{y} \in \mathbb{R}^m$ the vector of observations if $\mathbf{y} = \Phi \mathbf{x} + \mathbf{e}$ where $\mathbf{e} \in \mathbb{R}^m$ is a vector of noise or errors, possibly equal to $\mathbf{0}$. The task is to recover \mathbf{x} given only the observations \mathbf{y} and the matrix Φ . Clearly if $\mathbf{e} = \mathbf{0}$, $m = n$ and the columns of Φ are linearly independent, then Φ is invertible and \mathbf{x} can be recovered exactly. However compressive sensing asks (and answers) the question, how well can we do if $m \ll n$?

First, let us define the *Restricted Isometry Property* (RIP) condition of order s for a matrix $\Phi \in \mathbb{R}^{m \times n}$ as per [10]. Let $\delta_s \geq 0$ be the smallest value such that

$$1 - \delta_s \leq \frac{\|\Phi \mathbf{x}\|_2^2}{\|\mathbf{x}\|_2^2} \leq 1 + \delta_s \quad (1)$$

for all s -sparse vectors. We call δ_s the *restricted isometry constant of order s* . If $\delta_s < 1$ we say that Φ satisfies the RIP of order s . It is easy to see that we have

$$\lambda_{\min}(\Phi^\top \Phi) \leq 1 - \delta_s \leq 1 + \delta_s \leq \lambda_{\max}(\Phi^\top \Phi),$$

for any s . For a given matrix Φ the restricted isometry constant δ_s can be calculated by

$$\delta_s = \max_{\Gamma: |\Gamma|=s} \{1 - \lambda_{\min}(\Phi_\Gamma^\top \Phi_\Gamma), \lambda_{\max}(\Phi_\Gamma^\top \Phi_\Gamma) - 1\}.$$

This is proved in Lemma 5.

In order to give some sense to the problem of finding a vector \mathbf{x} , given the measurements \mathbf{y} so that $\mathbf{y} = \Phi \mathbf{x}$, we make the assumption that \mathbf{x} is sparse. We can then cast this question as a minimisation problem

$$\arg \min_{\hat{\mathbf{x}} \in \mathbb{R}^n} \|\hat{\mathbf{x}}\|_0 \text{ subject to } \mathbf{y} = \Phi \hat{\mathbf{x}}.$$

The following lemma from [12] gives conditions under which the minimiser to the ℓ_0 minimisation problem above is the same as the solution to $\mathbf{y} = \Phi \mathbf{x}$.

Lemma 1 (Lemma 1.2 of [12]). *Let $\Phi \in \mathbb{R}^{m \times n}$ be a matrix with RIP constant $\delta_{2s} < 1$ and let Γ be an index set with $|\Gamma| \leq s$. Let $\mathbf{x} \in \mathbb{R}^n$ be a vector with support Γ and set $\mathbf{y} = \Phi \mathbf{x}$. Then \mathbf{x} is the unique solution to*

$$\arg \min_{\hat{\mathbf{x}} \in \mathbb{R}^n} \|\hat{\mathbf{x}}\|_0 \quad \text{subject to} \quad \mathbf{y} = \Phi \hat{\mathbf{x}},$$

and hence \mathbf{x} can be reconstructed exactly from \mathbf{y} .

Lemma 1 only proves the existence of a solution but does not say anything about how to find it.

One question is, what matrices Φ satisfy the RIP of order s ? In general, it is not possible to design a matrix Φ that satisfies the RIP of a particular order. It has been shown that certain classes of random matrices will obey the RIP with very high probability. For example, let the columns of Φ be sampled uniformly at random from the unit sphere, or let each entry Φ_{ij} be sampled independently from the Gaussian distribution with mean 0 and variance $\frac{1}{m}$. Then Φ obeys the RIP of order s with probability at least $1 - \mathcal{O}(e^{-m})$ provided that $m \geq Cs \log(n/s)$ for some constant C . As a rule of thumb, for a Gaussian measurement matrix it suffices to take $m \approx 2s \log n$ so that ℓ_1 minimisation will work. [13, 3, 20]

2.2 Frames

Frames are a generalisation of bases for Hilbert spaces and have been heavily used in wavelet decompositions [14, 24]. Let \mathcal{H} be an m -dimensional Hilbert space and let $\Phi = \{\varphi_i \in \mathcal{H} : 1 \leq i \leq n\}$ be a set of n elements. Then Φ forms a frame for \mathcal{H} if there exist constants $0 < A \leq B < \infty$ such that

$$A \|\mathbf{x}\|_{\mathcal{H}}^2 \leq \sum_{i=1}^n |\langle \mathbf{x}, \varphi_i \rangle_{\mathcal{H}}|^2 \leq B \|\mathbf{x}\|_{\mathcal{H}}^2, \quad (2)$$

for all $x \in \mathcal{H}$ and where $\|\cdot\|_{\mathcal{H}}$ and $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ are the norm and scalar-product defined on the Hilbert space. The function $S : \mathcal{H} \rightarrow \mathcal{H}$ given by

$$Sx \triangleq \sum_j \langle x, \varphi_j \rangle_{\mathcal{H}} \varphi_j,$$

is called the *frame operator*. Then given the vector \mathbf{y} of observations where $y_i = \langle x, \varphi_i \rangle_{\mathcal{H}}$ we can reconstruct the original element $x \in \mathcal{H}$ via the iterative algorithm

$$x^{(i+1)} = x^{(i)} + \frac{2}{A+B} S(x - x^{(i)}), \quad (3)$$

where A and B are the frame bounds in (2) [24]. Note that although x is unknown, we know Sx from the measurement vector \mathbf{y} and hence this algorithm can be implemented since S is linear. This algorithm will converge from any starting element $x \in \mathcal{H}$. In particular if $x^{(0)} \equiv 0$, then the error satisfies

$$\|x - x^{(k)}\|_{\mathcal{H}} \leq \left(\frac{B-A}{B+A} \right)^k \|x\|_{\mathcal{H}}.$$

Faster algorithms also exist, which can be found, for example, in [23].

It was shown in [23] that using any positive value $\gamma \leq 2/(B+A)$ in place of $2/(B+A)$ will also result in correct reconstruction, but at a possibly slower rate of convergence.

This algorithm is identical to some of those used in iteratively solving least squares problems, such as *Richardson's first order method* [1, pp276-280].

2.3 Polynomial Acceleration

Iterative algorithms can often be sped up by considering the output of all the previous iterations, not just the very last iteration. *Semi-iterative* methods such as *polynomial acceleration* [21, 22] and *Richardson's second order method* [1, pp280-282] are two ways of doing this. The idea behind these two methods is to use the solution from previous iterations for example, if we use the update

$$\mathbf{x}^{(k)} = \sum_{i=0}^{k-1} c_{k,i} \mathbf{x}^{(i)}, \quad \sum_{i=0}^{k-1} c_{k,i} = 1.$$

Define the polynomial $P_k(t)$ by

$$P_k(t) \triangleq \sum_{i=0}^k c_{k,i} t^i, \quad P_k(1) = 1,$$

and it follows that the error equation is

$$\mathbf{x}^{(k)} - \mathbf{x} = P_k(B) (\mathbf{x}^{(0)} - \mathbf{x}),$$

where $P_k(B)$ is a polynomial in the matrix B , hence the name *polynomial acceleration*. The convergence rate is bounded above by

$$\max_{\lambda_{\min}(B) \leq t \leq \lambda_{\max}(B)} |P_k(t)|,$$

and so we wish to find the minimiser to

$$\min_{P_k: \deg P_k = k} \max_{t \in [0,1]} |P_k(t)|.$$

The minimising class of polynomials for this term is the Chebyshev polynomials of the first kind [31], defined by the recurrence relationship

$$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x),$$

with $T_0(x) = 1$ and $T_1(x) = x$. It is shown in [21] that we do not need to use all the previous iterates, as the polynomials are generated recursively it suffices to use

$$\mathbf{x}^{(1)} = \mathbf{x}^{(0)} + \gamma A^T (\mathbf{y} - A\mathbf{x}^{(0)})$$

and

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k-1)} + \omega^{(k+1)} \left(\gamma A^T (\mathbf{y} - A\mathbf{x}^{(k)}) + \mathbf{x}^{(k)} - \mathbf{x}^{(k-1)} \right),$$

where

$$\omega^{(k+1)} = \frac{1}{1 - \omega^{(k)} \frac{\mu^2}{4}}, \quad \omega^{(1)} = 1, \quad \mu = \frac{b-a}{b+a}$$

and where a, b are the minimum and maximum eigenvalues of $A^T A$. This particular method is known as the *Chebyshev semi-iterative method*. The second order Richardson method differs only slightly in that we take

$$\omega = \omega^{(k)} = \frac{2}{1 + \sqrt{1 - \mu^2}}.$$

In fact it can be shown that in the Chebyshev method we have $\omega^{(k)} \rightarrow \omega$ as $k \rightarrow \infty$. These algorithms typically converge an order of magnitude faster than without the acceleration [1]. In [23] it is also shown that this technique leads to significantly faster convergence of the frame reconstruction algorithm.

3 Reconstruction Algorithm

3.1 MFR – Modified Frame Reconstruction Algorithm

Our approach is based on the frame reconstruction algorithm in (3). The key observation for MFR is the fact that we can still perform this iterative step even if the matrix Ψ no longer forms a frame for the space \mathbb{R}^n . And that if the algorithm converges, it can still converge to a solution $\hat{\mathbf{x}}$ of the now under-determined matrix equation $\mathbf{y} = \Phi \mathbf{x}$, which is exactly the problem of compressive sensing.

We give the plain version of the MFR algorithm (Algorithm 1) and then show several modifications which increase both the rate of convergence and the success rate in finding the original sparse vector. We will also show theoretical bounds for the convergence and give sufficient conditions for convergence to occur.

The algorithm consists of two parts, an update and a thresholding step.

1. **Update:** Similar to the frame algorithm we perform an update

$$\mathbf{a}^{(k+1)} = \mathbf{x}^{(k)} + \gamma \Phi^T (\mathbf{y} - \Phi \mathbf{x}^{(k)}), \quad (4)$$

where \mathbf{y} is the vector of measurements, Φ is the measurement matrix and γ is a control parameter which we often refer to as the *step-length*.

2. **Thresholding:** The second part of the algorithm is the thresholding procedure where we generate the next “solution”

$$\mathbf{x}^{(k+1)} = \mathbb{H}_{\hat{s}} [\mathbf{a}^{(k+1)}]. \quad (5)$$

Here we simply threshold the output of the frame step producing an \hat{s} -sparse approximation.

Recall that $\mathbb{H}_{\hat{s}}(\mathbf{z})$ produces the best \hat{s} -sparse approximation to the input \mathbf{z} under any ℓ_p norm for $1 \leq p < \infty$. These two steps are repeated until “convergence” occurs, that is, the change from one iteration to the next is sufficiently small.

3.2 Analysis of Performance

We now state the properties of the MFR algorithm in Theorem 2 and Proposition 3. Proposition 3 gives sufficient convergence conditions in the scenario where we measure a sparse signal with noise, i.e. in the model $\mathbf{y} = \Phi \mathbf{x} + \mathbf{e}$ where $\Phi \in \mathbb{R}^{m \times n}$ is the measurement matrix, $\mathbf{x} \in \mathbb{R}^n$ is the s -sparse data, $\mathbf{e} \in \mathbb{R}^m$ is noise and $\mathbf{y} \in \mathbb{R}^m$ is the observed signal. We will see in Theorem 2 that the MFR algorithm converges even when the signal is not sparse. The proof of these results uses the same techniques as the proof in [6] although we choose to use the RIP condition rather than the modified RIP condition.

Theorem 2. Fix $\hat{s} = s$ where s is the sparsity of the desired solution. Then given measurements $\mathbf{y} = \Phi \mathbf{x} + \mathbf{e}$ where $\Phi \in \mathbb{R}^{m \times n}$ has the RIP such that either condition (a), (b) or (c) is satisfied

$$\begin{aligned} (a) \quad & \gamma \geq \frac{1}{\delta_{3s} - \delta_{2s} + 1}, \text{ and } \gamma \delta_{3s} \leq \frac{1}{\sqrt{32}}, \text{ or} \\ (b) \quad & \gamma < \frac{1}{\delta_{3s} - \delta_{2s} + 1}, \text{ and } \gamma(1 - \delta_{2s}) \geq 1 - \frac{1}{\sqrt{32}}, \text{ or} \\ (c) \quad & \frac{3}{4(1 - \delta_{2s})} < \gamma < \frac{1}{1 - \delta_{2s}} \text{ and } \delta_{2s} < 1, \end{aligned}$$

Algorithm 1 Modified Frame Reconstruction Algorithm

Input:

- The measurement matrix Φ .
- Observation vector \mathbf{y} .
- Estimate of sparsity \hat{s} of the vector \mathbf{x} .
- Step size γ .
- Tolerance parameter ε_{tol} .

Output:

- A vector $\hat{\mathbf{x}} \in \mathbb{R}^n$ that is \hat{s} -sparse.

```
1:  $\mathbf{x}^{(0)} \leftarrow \mathbf{0}$ 
2:  $k \leftarrow 1$ 
3: while  $\|\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}\|_2 \geq \varepsilon_{tol}$  do
4:    $\mathbf{x}^{(k+1)} \leftarrow \mathbb{H}_{\hat{s}}[\mathbf{x}^{(k)} + \gamma \Phi^T(\mathbf{y} - \Phi \mathbf{x}^{(k)})]$ 
5:    $k \leftarrow k + 1$ 
6: end while
7: return  $\mathbf{x}^{(k)}$ 
```

then Algorithm 1 will recover an approximation $\mathbf{x}^{(k)}$ satisfying

$$\begin{aligned} \|\mathbf{x}^{(k)} - \mathbf{x}^s\|_2 &\leq 2^{-k} \|\mathbf{x}^s\|_2 + 4\gamma \sqrt{1 + \delta_{2s}} \|\mathbf{e}\|_2 + \dots \\ &\quad 4\gamma(1 + \delta_{2s}) \left(\|\mathbf{x} - \mathbf{x}^{2s}\|_2 + \frac{1}{\sqrt{2s}} \|\mathbf{x} - \mathbf{x}^{2s}\|_1 \right), \end{aligned} \quad (7)$$

where \mathbf{x}^{2s} is the best $2s$ -sparse approximation to \mathbf{x} .

Proposition 3. Under the conditions of Theorem 2 and given measurements $\mathbf{y} = \Phi \mathbf{x} + \mathbf{e}$ where \mathbf{x} is s -sparse for $s \leq \hat{s}$ and $\Phi \in \mathbb{R}^{m \times n}$ has the RIP such that either condition (a), (b) or (c) is satisfied

$$\begin{aligned} (a) \quad &\gamma \geq \frac{1}{\delta_{s+2\hat{s}} - \delta_{s+\hat{s}} + 1}, \text{ and } \gamma \delta_{s+2\hat{s}} \leq \frac{1}{\sqrt{32}}, \text{ or} \\ (b) \quad &\gamma < \frac{1}{\delta_{s+2\hat{s}} - \delta_{s+\hat{s}} + 1}, \text{ and } \gamma(1 - \delta_{s+\hat{s}}) \geq 1 - \frac{1}{\sqrt{32}}, \text{ or} \\ (c) \quad &\frac{3}{4(1 - \delta_{s+\hat{s}})} < \gamma < \frac{1}{1 - \delta_{s+\hat{s}}} \text{ and } \delta_{s+\hat{s}} < 1, \end{aligned}$$

then Algorithm 1 will recover an approximation $\mathbf{x}^{(k)}$ satisfying

$$\|\mathbf{x} - \mathbf{x}^{(k)}\|_2 \leq 2^{-k} \|\mathbf{x}\|_2 + 4\gamma \sqrt{1 + \delta_{s+\hat{s}}} \|\mathbf{e}\|_2. \quad (8)$$

Proof. Conditions (a) and (b) come from Lemma 4 and condition (c) follows by setting $\alpha = \frac{1}{2}$ in Lemma 8. \square

We will first prove Lemmas 4 and 8, hence proving Proposition 3 and then use this to prove the main theorem.

Lemma 4. *Under the conditions of Theorem 2 and given measurements $\mathbf{y} = \Phi\mathbf{x} + \mathbf{e}$ where \mathbf{x} is s -sparse for $s \leq \hat{s}$ and $\Phi \in \mathbb{R}^{m \times n}$ has the RIP such that either condition (a) or condition (b) is satisfied*

$$(a) \quad \gamma \geq \frac{1}{\delta_{s+2\hat{s}} - \delta_{s+\hat{s}} + 1}, \text{ and } \gamma\delta_{s+2\hat{s}} \leq \frac{1}{\sqrt{32}}, \quad \text{or}$$

$$(b) \quad \gamma < \frac{1}{\delta_{s+2\hat{s}} - \delta_{s+\hat{s}} + 1}, \text{ and } \gamma(1 - \delta_{s+\hat{s}}) \geq 1 - \frac{1}{\sqrt{32}},$$

then Algorithm 1 will recover an approximation $\mathbf{x}^{(k)}$ satisfying

$$\|\mathbf{x} - \mathbf{x}^{(k)}\|_2 \leq 2^{-k} \|\mathbf{x}\|_2 + 4\gamma\sqrt{1 + \delta_{s+\hat{s}}} \|\mathbf{e}\|_2. \quad (10)$$

To prove Lemma 4, we need the following lemmas.

Lemma 5. *Suppose that $\Phi \in \mathbb{R}^{m \times n}$ obeys the restricted isometry property of order s with value δ_s . Then for any set of indices Γ such that $|\Gamma| \leq s$ the singular values of Φ_Γ lie in the range $[\sqrt{1 - \delta_s}, \sqrt{1 + \delta_s}]$. Furthermore the eigenvalues of $\Phi_\Gamma^\top \Phi_\Gamma - \mathbf{I}$ lie in the interval $[-\delta_s, \delta_s]$.*

Proof. Recall that the RIP says that for all vectors $\mathbf{x}' \in \mathbb{R}^t$ for $t \leq s$

$$1 - \delta_s \leq \frac{\|\Phi_\Gamma \mathbf{x}'\|_2^2}{\|\mathbf{x}'\|_2^2} \leq 1 + \delta_s.$$

Furthermore we know that the Rayleigh quotient for a matrix A is bounded by the minimum and maximum eigenvalues of A , i.e. $\lambda_{\min}(A) \leq \rho_A(\mathbf{x}) \leq \lambda_{\max}(A)$ for all \mathbf{x} . Set $A = \Phi_\Gamma^\top \Phi_\Gamma$, then the eigenvalues of A are the square of the singular values of Φ_Γ . As the eigenvalues for A are bounded by $1 \pm \delta_s$ we have the corresponding bound for the singular values of Φ_Γ , namely $\sigma(\Phi_\Gamma) \in [\sqrt{1 - \delta_s}, \sqrt{1 + \delta_s}]$. It follows then that the eigenvalues of $\Phi_\Gamma^\top \Phi_\Gamma - \mathbf{I}$ lie in the interval $[-\delta_s, \delta_s]$. \square

Lemma 6. *For all index sets Γ and all measurement matrices Φ for which the RIP holds with $s = |\Gamma|$*

$$\|(\mathbf{I} - \Phi_\Gamma^\top \Phi_\Gamma) \mathbf{x}_\Gamma\|_2 \leq \delta_s \|\mathbf{x}_\Gamma\|_2. \quad (11)$$

and furthermore

$$\|(\mathbf{I} - \gamma\Phi_\Gamma^\top \Phi_\Gamma) \mathbf{x}_\Gamma\|_2 \leq [1 - \gamma(1 - \delta_s)] \|\mathbf{x}_\Gamma\|_2. \quad (12)$$

Proof. The RIP guarantees us that the eigenvalues of $\Phi_\Gamma^\top \Phi_\Gamma$ lie in the range $1 - \delta_s$ to $1 + \delta_s$. Hence the matrix $\mathbf{I} - \Phi_\Gamma^\top \Phi_\Gamma$ has eigenvalues in $[-\delta_s, \delta_s]$. For the second result we proceed similarly. Clearly $\gamma\Phi_\Gamma^\top \Phi_\Gamma$ has eigenvalues in the range $[\gamma(1 - \delta_s), \gamma(1 + \delta_s)]$, hence the maximum eigenvalue of $\mathbf{I} - \gamma\Phi_\Gamma^\top \Phi_\Gamma$ is less than or equal to $1 - \gamma(1 - \delta_s)$.

Then for each fixed Γ we have

$$\begin{aligned} \|(\mathbf{I} - \gamma\Phi_\Gamma^\top \Phi_\Gamma) \mathbf{x}_\Gamma\|_2 &\leq \|(\mathbf{I} - \gamma\Phi_\Gamma^\top \Phi_\Gamma)\|_2 \cdot \|\mathbf{x}_\Gamma\|_2 \\ &\leq (1 - \gamma(1 - \delta_s)) \|\mathbf{x}_\Gamma\|_2, \end{aligned}$$

and the first equation follows by setting $\gamma = 1$. \square

Lemma 7 (Based on Lemma 2 of [6]). *Let Γ and Λ be two disjoint index sets for the matrix Φ . Then for all Φ for which the RIP holds with $s = |\Gamma \cup \Lambda|$*

$$\|\Phi_\Gamma^\top \Phi_\Lambda \mathbf{x}_\Lambda\|_2 \leq \delta_s \|\mathbf{x}_\Lambda\|_2. \quad (14)$$

Proof. Set $\Omega = \Gamma \cup \Lambda$. Since Γ and Λ are disjoint, the matrix $(-\Phi_\Gamma^\top \Phi_\Lambda)$ is a submatrix of the matrix $\mathbf{I} - \Phi_\Omega^\top \Phi_\Omega$. Since the largest singular value of a submatrix is bounded above by the largest singular value of the full matrix, we have

$$\|\Phi_\Gamma^\top \Phi_\Lambda\|_2 \leq \|\mathbf{I} - \Phi_\Omega^\top \Phi_\Omega\|_2 \leq \delta_s$$

as the eigenvalues of $\mathbf{I} - \Phi_\Omega^\top \Phi_\Omega$ are bounded above by δ_s . Hence

$$\begin{aligned} \|\Phi_\Gamma^\top \Phi_\Lambda \mathbf{x}_\Lambda\|_2 &\leq \|\Phi_\Gamma^\top \Phi_\Lambda\|_2 \cdot \|\mathbf{x}_\Lambda\|_2 \\ &\leq \delta_s \|\mathbf{x}_\Lambda\|_2, \end{aligned}$$

completing the lemma. \square

We now have the necessary results to prove Lemma 4.

Proof of Lemma 4. Note that this proof follows parts of the proof in [6]. First put

$$\begin{aligned} \mathbf{r}^{(i)} &\triangleq \mathbf{x} - \mathbf{x}^{(i)}, \\ \mathbf{a}^{(i)} &\triangleq \mathbf{x}^{(i-1)} + \gamma \Phi^\top (\mathbf{y} - \Phi \mathbf{x}^{(i-1)}), \\ \mathbf{x}^{(i)} &\triangleq \mathbb{H}_s(\mathbf{a}^{(i)}), \\ \Gamma^* &\triangleq \text{supp}(\mathbf{x}), \\ \Gamma^{(i)} &\triangleq \text{supp}(\mathbf{x}^{(i)}), \\ B^{(i)} &\triangleq \Gamma^* \cup \Gamma^{(i)}. \end{aligned}$$

As a consequence we have $|\Gamma^*| \leq s$ and $|\Gamma^{(i)}| \leq \hat{s}$.

Consider the error $\|\mathbf{x}^s - \mathbf{x}^{(i+1)}\|_2$. Now we have

$$\begin{aligned} \mathbf{x} &= \mathbf{x}^s = \mathbf{x}_{\Gamma^*} = \mathbf{x}_{B^{(i)}} \quad \text{and} \\ \mathbf{x}^{(i)} &= \mathbf{x}_{\Gamma^{(i)}} = \mathbf{x}_{B^{(i)}}^{(i)}. \end{aligned}$$

Applying the triangle inequality we get

$$\|\mathbf{x}^s - \mathbf{x}^{(i+1)}\|_2 \leq \|\mathbf{x}_{B^{(i+1)}}^s - \mathbf{a}_{B^{(i+1)}}^{(i+1)}\|_2 + \|\mathbf{x}_{B^{(i+1)}}^{(i+1)} - \mathbf{a}_{B^{(i+1)}}^{(i+1)}\|_2. \quad (18)$$

As $\mathbf{x}^{(i+1)}$ is the thresholded version of $\mathbf{a}^{(i+1)}$ it is the best s -term approximation to $\mathbf{a}^{(i+1)}$, in particular it is better than \mathbf{x}^s . Hence

$$\|\mathbf{x}^{(i+1)} - \mathbf{a}_{B^{(i+1)}}^{(i+1)}\|_2 \leq \|\mathbf{x}^s - \mathbf{a}_{B^{(i+1)}}^{(i+1)}\|_2,$$

and thus (18) becomes

$$\|\mathbf{x}^s - \mathbf{x}^{(i+1)}\|_2 \leq 2 \|\mathbf{x}_{B^{(i+1)}}^s - \mathbf{a}_{B^{(i+1)}}^{(i+1)}\|_2.$$

Using the fact that $\mathbf{y} = \Phi \mathbf{x}^s + \mathbf{e}$ and $\mathbf{r}^{(i)} = \mathbf{x}^s - \mathbf{x}^{(i)}$ we get

$$\begin{aligned} \mathbf{a}_{B^{(i+1)}}^{(i+1)} &= \mathbf{x}_{B^{(i+1)}}^{(i)} + \gamma \Phi_{B^{(i+1)}}^\top \left(\mathbf{y} - \Phi \mathbf{x}_{B^{(i+1)}}^{(i)} \right) \\ &= \mathbf{x}_{B^{(i+1)}}^{(i)} + \gamma \Phi_{B^{(i+1)}}^\top \Phi \mathbf{r}^{(i)} + \gamma \Phi_{B^{(i+1)}}^\top \mathbf{e}, \end{aligned}$$

hence

$$\begin{aligned} \left\| \mathbf{x}^s - \mathbf{x}^{(i+1)} \right\|_2 &\leq 2 \left\| \underbrace{\mathbf{x}_{B^{(i+1)}}^s - \mathbf{x}_{B^{(i+1)}}^{(i)}}_{\mathbf{r}_{B^{(i+1)}}^{(i)}} - \gamma \Phi_{B^{(i+1)}}^\top \Phi \mathbf{r}^{(i)} - \gamma \Phi_{B^{(i+1)}}^\top \mathbf{e} \right\|_2 \\ &\leq 2 \left\| \mathbf{r}_{B^{(i+1)}}^{(i)} - \gamma \Phi_{B^{(i+1)}}^\top \Phi \mathbf{r}^{(i)} \right\|_2 + 2\gamma \left\| \Phi_{B^{(i+1)}}^\top \mathbf{e} \right\|_2 \\ &= 2 \left\| \left(\mathbf{I} - \gamma \Phi_{B^{(i+1)}}^\top \Phi_{B^{(i+1)}} \right) \mathbf{r}_{B^{(i+1)}}^{(i)} - \dots \right. \\ &\quad \left. - \gamma \Phi_{B^{(i+1)}}^\top \Phi_{B^{(i)} \setminus B^{(i+1)}} \mathbf{r}_{B^{(i)} \setminus B^{(i+1)}}^{(i)} \right\|_2 + 2\gamma \left\| \Phi_{B^{(i+1)}}^\top \mathbf{e} \right\|_2 \\ &\leq 2 \left\| \left(\mathbf{I} - \gamma \Phi_{B^{(i+1)}}^\top \Phi_{B^{(i+1)}} \right) \mathbf{r}_{B^{(i+1)}}^{(i)} \right\|_2 + \dots \\ &\quad + 2\gamma \left\| \Phi_{B^{(i+1)}}^\top \Phi_{B^{(i)} \setminus B^{(i+1)}} \mathbf{r}_{B^{(i)} \setminus B^{(i+1)}}^{(i)} \right\|_2 + 2\gamma \left\| \Phi_{B^{(i+1)}}^\top \mathbf{e} \right\|_2, \end{aligned} \quad (20)$$

by repeated application of the triangle inequality and by splitting the residual into two parts, $\mathbf{r}^{(i+1)} = \mathbf{r}_{B^{(i+1)}}^{(i+1)} + \mathbf{r}_{B^{(i+1)} \setminus B^{(i)}}^{(i+1)}$. Then

$$\left| B^{(i)} \cup B^{(i+1)} \right| = \left| \Gamma^* \cup \Gamma^{(i)} \cup \Gamma^{(i+1)} \right| \leq s + 2\hat{s}$$

as each set $\Gamma^{(k)}$ has at most \hat{s} entries and $|\Gamma^*| \leq s$. Recall from the RIP and Lemmas 6 and 7 that

$$\left\| \Phi_\Lambda^\top \mathbf{x} \right\|_2^2 \leq (1 + \delta_s) \left\| \mathbf{x} \right\|_2^2, \quad (21a)$$

$$\left\| \left(\mathbf{I} - \gamma \Phi_\Lambda^\top \Phi_\Lambda \right) \mathbf{x}_\Lambda \right\|_2 \leq (1 - \gamma(1 - \delta_s)) \left\| \mathbf{x}_\Lambda \right\|_2, \quad (21b)$$

$$\left\| \Phi_\Omega^\top \Phi_{\Omega'} \mathbf{x}_{\Omega'} \right\|_2 \leq \delta_s \left\| \mathbf{x}_{\Omega'} \right\|_2, \quad (21c)$$

for all matrices Φ which obey the RIP and sets Λ, Ω, Ω' , where Ω, Ω' are disjoint, $|\Lambda| = s$ and $|\Omega \cup \Omega'| = s$. We also have $\delta_s \leq \delta_{s'}$ for all positive integers $s \leq s'$. Applying (21b) to the first term in (20), and applying (21c) and (21a) to the second and third terms respectively, we get

$$\begin{aligned} \left\| \mathbf{r}^{(i+1)} \right\|_2 &\leq 2(1 - \gamma(1 - \delta_{s+\hat{s}})) \left\| \mathbf{r}_{B^{(i+1)}}^{(i)} \right\|_2 + \dots \\ &\quad + 2\gamma\delta_{s+2\hat{s}} \left\| \mathbf{r}_{B^{(i)} \setminus B^{(i+1)}}^{(i)} \right\|_2 + \dots \\ &\quad + 2\gamma\sqrt{1 + \delta_{s+\hat{s}}} \left\| \mathbf{e} \right\|_2. \end{aligned} \quad (22)$$

The vectors $\mathbf{r}_{B^{(i+1)}}^{(i)}$ and $\mathbf{r}_{B^{(i)} \setminus B^{(i+1)}}^{(i)}$ are orthogonal as they have disjoint supports. Now let $\mathbf{u}, \mathbf{v} \in \mathbb{R}^n$ be two orthogonal vectors, then

$$\left\| \mathbf{u} \right\|_2 + \left\| \mathbf{v} \right\|_2 \leq \sqrt{2} \left\| \mathbf{u} + \mathbf{v} \right\|_2.$$

We use this to bound the sum of the two terms $\left\| \mathbf{r}_{B^{(i+1)}}^{(i)} \right\|_2$ and $\left\| \mathbf{r}_{B^{(i)} \setminus B^{(i+1)}}^{(i)} \right\|_2$.

We first ask, how do the terms $1 - \gamma(1 - \delta_{s+\hat{s}})$ and $\gamma\delta_{s+2\hat{s}}$ compare, given that $\delta_{s+\hat{s}} \leq \delta_{s+2\hat{s}}$? We then have either

$$1 - \gamma + \gamma\delta_{s+\hat{s}} \leq \gamma\delta_{s+2\hat{s}} \iff \gamma \geq \frac{1}{\delta_{s+2\hat{s}} - \delta_{s+\hat{s}} + 1}, \quad \text{or} \quad (23a)$$

$$1 - \gamma + \gamma\delta_{s+\hat{s}} > \gamma\delta_{s+2\hat{s}} \iff \gamma < \frac{1}{\delta_{s+2\hat{s}} - \delta_{s+\hat{s}} + 1}. \quad (23b)$$

- **Case 1 – Equation (23a):** Equation (22) becomes

$$\|\mathbf{r}^{(i+1)}\|_2 \leq 2\sqrt{2}\gamma\delta_{s+2\hat{s}} \|\mathbf{r}^{(i)}\|_2 + 2\gamma\sqrt{1 + \delta_{s+\hat{s}}} \|\mathbf{e}\|_2.$$

Then if

$$2\sqrt{2}\gamma\delta_{s+2\hat{s}} \leq \frac{1}{2} \iff \gamma\delta_{s+2\hat{s}} \leq \frac{1}{\sqrt{32}},$$

we have

$$\|\mathbf{r}^{(i+1)}\|_2 \leq \frac{1}{2} \|\mathbf{r}^{(i)}\|_2 + 2\gamma\sqrt{1 + \delta_{s+\hat{s}}} \|\mathbf{e}\|_2.$$

Hence

$$\|\mathbf{x} - \mathbf{x}^{(k)}\|_2 \leq 2^{-k} \|\mathbf{x}\|_2 + 4\gamma\sqrt{1 + \delta_{s+\hat{s}}} \|\mathbf{e}\|_2,$$

provided that

$$\gamma \geq \frac{1}{\delta_{s+2\hat{s}} - \delta_{s+\hat{s}} + 1}, \quad \text{and}$$

$$\gamma\delta_{s+2\hat{s}} \leq \frac{1}{\sqrt{32}} \approx 0.177.$$

- **Case 2 – Equation (23b):** Equation (22) becomes

$$\|\mathbf{r}^{(i+1)}\|_2 \leq 2\sqrt{2}(1 - \gamma(1 - \delta_{s+\hat{s}})) \|\mathbf{r}^{(i)}\|_2 + 2\gamma\sqrt{1 + \delta_{s+\hat{s}}} \|\mathbf{e}\|_2.$$

If

$$2\sqrt{2}(1 - \gamma(1 - \delta_{s+\hat{s}})) \leq \frac{1}{2} \iff \gamma(1 - \delta_{s+\hat{s}}) \geq 1 - \frac{1}{\sqrt{32}}$$

$$\iff \delta_{s+\hat{s}} \leq 1 - \frac{1}{\gamma} + \frac{1}{\gamma\sqrt{32}} = \frac{8\gamma - 8 + \sqrt{2}}{8\gamma},$$

we again have

$$\|\mathbf{r}^{(i+1)}\|_2 \leq \frac{1}{2} \|\mathbf{r}^{(i)}\|_2 + 2\gamma\sqrt{1 + \delta_{s+\hat{s}}} \|\mathbf{e}\|_2.$$

Hence

$$\|\mathbf{x} - \mathbf{x}^{(k)}\|_2 \leq 2^{-k} \|\mathbf{x}\|_2 + 4\gamma\sqrt{1 + \delta_{s+\hat{s}}} \|\mathbf{e}\|_2,$$

provided that

$$\gamma < \frac{1}{\delta_{s+2\hat{s}} - \delta_{s+\hat{s}} + 1}, \quad \text{and}$$

$$\gamma(1 - \delta_{s+\hat{s}}) \geq 1 - \frac{1}{\sqrt{32}} \approx 0.82,$$

Putting these two results together we have

$$\left\| \mathbf{r}^{(i+1)} \right\|_2 \leq \frac{1}{2} \left\| \mathbf{r}^{(i)} \right\|_2 + 2\gamma \sqrt{1 + \delta_{s+\hat{s}}} \|\mathbf{e}\|_2,$$

if either of the following conditions (a) or (b) are met

$$(a) \quad \gamma \geq \frac{1}{\delta_{s+2\hat{s}} - \delta_{s+\hat{s}} + 1}, \quad \text{and} \quad \gamma \delta_{s+2\hat{s}} \leq \frac{1}{\sqrt{32}}, \quad \text{or} \quad (27a)$$

$$(b) \quad \gamma < \frac{1}{\delta_{s+2\hat{s}} - \delta_{s+\hat{s}} + 1}, \quad \text{and} \quad \gamma(1 - \delta_{s+\hat{s}}) \geq 1 - \frac{1}{\sqrt{32}}, \quad (27b)$$

completing the proof of the lemma. \square

Looking at our algorithm in another way shows that the MFR algorithm is capable of attaining the bounds of Lemma 1. Lemma 1 says that given $\mathbf{y} = \Phi \mathbf{x}$, then the minimiser to $\|\hat{\mathbf{x}}\|_0$ subject to $\mathbf{y} = \Phi \hat{\mathbf{x}}$ is unique and equal to \mathbf{x} if $\delta_{2s} < 1$. Lemma 8 and its corollary shows that the MFR algorithm can achieve this bound.

Lemma 8. *Let $\mathbf{x} \in \mathbb{R}^{m \times n}$ be an s -sparse vector and let the matrix $\Phi \in \mathbb{R}^{m \times n}$ have RIP constants δ_s and assume we are given the measurements $\mathbf{y} = \Phi \mathbf{x} + \mathbf{e}$. If*

$$\frac{1 - \frac{\alpha}{2}}{1 - \delta_{s+\hat{s}}} < \gamma < \frac{1}{1 - \delta_{s+\hat{s}}}, \quad (28)$$

for some constant $0 < \alpha < 1$ and $\hat{s} \geq s$, and

$$\delta_{s+\hat{s}} < 1, \quad (29)$$

then the MFR algorithm produces an approximation $\mathbf{x}^{(k)}$ satisfying

$$\left\| \mathbf{x} - \mathbf{x}^{(k)} \right\|_2 \leq \alpha^k \|\mathbf{x}\|_2 + \frac{2\gamma \sqrt{1 + \delta_{s+\hat{s}}}}{1 - \alpha} \|\mathbf{e}\|_2. \quad (30)$$

Proof. Recall the fundamental step of the MFR algorithm

$$\mathbf{x}^{(k+1)} = \mathbb{H}_{\hat{s}} \left(\mathbf{x}^{(k)} + \gamma \Phi^T \left(\mathbf{y} - \Phi \mathbf{x}^{(k)} \right) \right). \quad (31)$$

As before, set $B^{(k)} \triangleq \text{supp}(\mathbf{x}) \cup \text{supp}(\mathbf{x}^{(k)})$ and $\mathbf{a}^{(k+1)} \triangleq \mathbf{x}^{(k)} + \gamma \Phi^T (\mathbf{y} - \Phi \mathbf{x}^{(k)})$. Then we have

$$\begin{aligned} \mathbf{a}_{B^{(k+1)}}^{(k+1)} &= \mathbf{x}_{B^{(k+1)}}^{(k)} + \gamma \Phi_{B^{(k+1)}}^T \left(\mathbf{y} - \Phi \mathbf{x}_{B^{(k+1)}}^{(k)} \right) \\ &= \mathbf{x}_{B^{(k+1)}}^{(k)} + \gamma \Phi_{B^{(k+1)}}^T \Phi (\mathbf{x} - \mathbf{x}^{(k)}) + \gamma \Phi_{B^{(k+1)}}^T \mathbf{e}. \end{aligned}$$

This gives the error estimate

$$\begin{aligned} \left\| \mathbf{x} - \mathbf{x}^{(k+1)} \right\|_2 &\leq \left\| \mathbf{x} - \mathbf{a}_{B^{(k+1)}}^{(k+1)} \right\|_2 + \left\| \mathbf{a}_{B^{(k+1)}}^{(k+1)} - \mathbf{x}^{(k+1)} \right\|_2 \\ &\leq 2 \left\| \mathbf{x} - \mathbf{a}_{B^{(k+1)}}^{(k+1)} \right\|_2 \\ &= 2 \left\| \mathbf{x} - \mathbf{x}_{B^{(k+1)}}^{(k)} - \gamma \Phi_{B^{(k+1)}}^T \Phi (\mathbf{x} - \mathbf{x}_{B^{(k+1)}}^{(k)}) - \gamma \Phi_{B^{(k+1)}}^T \mathbf{e} \right\|_2 \\ &\leq 2 \left\| \mathbf{x} - \mathbf{x}_{B^{(k+1)}}^{(k)} - \gamma \Phi_{B^{(k+1)}}^T \Phi (\mathbf{x} - \mathbf{x}_{B^{(k+1)}}^{(k)}) \right\|_2 + 2 \|\gamma \Phi^T \mathbf{e}\|_2 \\ &= 2 \left\| (\mathbf{I} - \gamma \Phi^T \Phi) (\mathbf{x} - \mathbf{x}^{(k)}) \right\|_2 + 2\gamma \|\Phi_{B^{(k+1)}}^T \mathbf{e}\|_2 \\ &\leq 2(1 - \gamma(1 - \delta_{s+\hat{s}})) \left\| \mathbf{x} - \mathbf{x}^{(k)} \right\|_2 + 2\gamma \sqrt{1 + \delta_{s+\hat{s}}} \|\mathbf{e}\|_2, \end{aligned}$$

by Lemmas 6 and 7. This implies that

$$\|\mathbf{x} - \mathbf{x}^{(k)}\|_2 \leq \left[2\left(1 - \gamma(1 - \delta_{s+\hat{s}})\right)\right]^k \|\mathbf{x}\|_2 + \frac{2\gamma\sqrt{1 + \delta_{s+\hat{s}}}}{1 - 2\left(1 - \gamma(1 - \delta_{s+\hat{s}})\right)} \|\mathbf{e}\|_2,$$

since $\mathbf{x}^{(0)} = \mathbf{0}$. Thus if $0 < 2(1 - \gamma(1 - \delta_{s+\hat{s}})) \leq \alpha < 1$ then the algorithm will converge. Since

$$\begin{aligned} 0 < 2(1 - \gamma(1 - \delta_{s+\hat{s}})) \leq \alpha &\iff 1 - \frac{\alpha}{2} \leq \gamma(1 - \delta_{s+\hat{s}}) < 1 - \delta_{s+\hat{s}} \\ &\iff \frac{1 - \frac{\alpha}{2}}{1 - \delta_{s+\hat{s}}} \leq \gamma < \frac{1}{1 - \delta_{s+\hat{s}}}, \end{aligned}$$

the algorithm will converge provided $\delta_{s+\hat{s}} < 1$ producing an approximation that obeys

$$\|\mathbf{x} - \mathbf{x}^{(k)}\|_2 \leq \alpha^k \|\mathbf{x}\|_2 + \frac{2\gamma\sqrt{1 + \delta_{s+\hat{s}}}}{1 - \alpha} \|\mathbf{e}\|_2,$$

completing the lemma. \square

This lemma says that for the right value of γ and provided that $\delta_{s+\hat{s}} < 1$, the algorithm will always converge to the correct solution, at the cost of noise amplification.

Corollary 9. *Under the hypothesis of Lemma 8 and if we could measure the signal \mathbf{y} exactly, i.e. if $\mathbf{e} = \mathbf{0}$, then for some value of γ , setting $\hat{s} = s$ gives an algorithm capable of attaining the bound in Lemma 1.*

Proof. Setting $\hat{s} = s$ in (29) gives the condition $\delta_{2s} < 1$ and by choosing γ so that

$$\frac{1 - \frac{\alpha}{2}}{1 - \delta_{2s}} \leq \gamma < \frac{1}{1 - \delta_{2s}},$$

then the algorithm will produce a sequence of approximations which converge to \mathbf{x} . \square

What is interesting about these results, is that they rely only on the *lower* RIP constant, that is, it only requires

$$0 < 1 - \delta_{2s} \leq \frac{\|\Phi\mathbf{x}\|_2^2}{\|\mathbf{x}\|_2^2},$$

for all $2s$ -sparse vectors \mathbf{x} . This means that the MFR algorithm will recover the correct s -sparse solution \mathbf{x} to $\mathbf{y} = \Phi\mathbf{x}$ provided that there are no $2s$ -sparse vectors in the kernel of Φ . In fact this is a tight theoretical bound: assume that there exists a vector $\mathbf{v} \in \mathbb{R}^n$ that is $2s$ -sparse and $\Phi\mathbf{v} = \mathbf{0}$. Choose a set $\Gamma \subset \{1, 2, \dots, n\}$ of size s so that $\Gamma \subset \text{supp}(\mathbf{v})$ and set $\mathbf{x} = -\mathbf{v}_\Gamma$ so that \mathbf{x} is s -sparse. Then

$$\mathbf{y} = \Phi\mathbf{x} = \Phi\mathbf{x} + \Phi\mathbf{v} = \Phi(\mathbf{x} + \mathbf{v}) = \Phi\mathbf{u},$$

where $\mathbf{u} = \mathbf{x} + \mathbf{v}$ is s -sparse and $\mathbf{u} \neq \mathbf{x}$. Hence there is no unique minimiser to $\|\hat{\mathbf{x}}\|_0$ subject to $\mathbf{y} = \Phi\hat{\mathbf{x}}$ and no algorithm will be able to return the correct solution 100% of the time. Thus the MFR algorithm is able to achieve the theoretically maximum performance.

Perhaps even more surprising is that the algorithm will converge as fast as we want (but still linearly), i.e. $\|\mathbf{x} - \mathbf{x}^{(k)}\|_2 \leq \alpha^k \|\mathbf{x}\|_2$ for any $0 < \alpha < 1$, provided we can choose γ so that $1 - \frac{\alpha}{2} \leq \gamma(1 - \delta_{2s}) < 1$.

This seems to be an astounding result, until we realise that this requires accurate values of δ_{2s} and explicitly calculating δ_{2s} for a random matrix is computationally equivalent to directly solving the ℓ_0 minimisation problem $\arg \min \|\hat{\mathbf{x}}\|_0$ subject to $\mathbf{y} = \Phi\hat{\mathbf{x}}$.

We introduce one final lemma before proving the main theorem.

Lemma 10 (Reduction to sparse case, Lemma 6.1 of [27]). *Let \mathbf{x} be a vector from \mathbb{R}^n and assume that Φ obeys the RIP of order t , then the sample vector $\mathbf{y} = \Phi\mathbf{x} + \mathbf{e}$ can also be written as $\mathbf{y} = \Phi\mathbf{x}^t + \tilde{\mathbf{e}}$ where*

$$\|\tilde{\mathbf{e}}\|_2 \leq \sqrt{1 + \delta_t} \left(\|\mathbf{x} - \mathbf{x}^t\|_2 + \frac{1}{\sqrt{t}} \|\mathbf{x} - \mathbf{x}^t\|_1 \right) + \|\mathbf{e}\|_2, \quad (35)$$

for any $t \in \mathbb{Z}$.

We now use Proposition 3 and Lemma 10 from [27] to prove the main theorem. This uses the same techniques as in [6].

Proof of Theorem 2. Let \mathbf{x}^s be the best s -sparse approximation to \mathbf{x} . Then observe that

$$\|\mathbf{x} - \mathbf{x}^{(k)}\|_2 \leq \|\mathbf{x} - \mathbf{x}^s\|_2 + \|\mathbf{x}^{(k)} - \mathbf{x}^s\|_2.$$

We then apply the algorithm to \mathbf{y} to recover an s -sparse approximation. From Proposition 3 we get the bound

$$\|\mathbf{x}^{(k)} - \mathbf{x}^s\|_2 \leq 2^{-k} \|\mathbf{x}^s\|_2 + 4\gamma\sqrt{1 + \delta_{2s}} \|\hat{\mathbf{e}}\|_2,$$

where $\hat{\mathbf{e}} = \mathbf{y} - \Phi\mathbf{x}^s$. Using Lemma 10 and setting $t = 2s$ we can write $\mathbf{y} = \Phi\mathbf{x}^{2s} + \tilde{\mathbf{e}}$ where \mathbf{x}^{2s} is a best $2s$ -sparse approximation to \mathbf{x} , such that

$$\|\tilde{\mathbf{e}}\|_2 \leq \sqrt{1 + \delta_{2s}} \left(\|\mathbf{x} - \mathbf{x}^{2s}\|_2 + \frac{1}{\sqrt{2s}} \|\mathbf{x} - \mathbf{x}^{2s}\|_1 \right) + \|\mathbf{e}\|_2,$$

Hence combining Lemma 10 and Proposition 3 we get

$$\begin{aligned} \|\mathbf{x}^{(k)} - \mathbf{x}^s\|_2 &\leq 2^{-k} \|\mathbf{x}^s\|_2 + 4\gamma\sqrt{1 + \delta_{2s}} \|\mathbf{e}\|_2 + \dots \\ &\quad + 4\gamma(1 + \delta_{2s}) \left(\|\mathbf{x} - \mathbf{x}^{2s}\|_2 + \frac{1}{\sqrt{2s}} \|\mathbf{x} - \mathbf{x}^{2s}\|_1 \right), \end{aligned}$$

under conditions (a), (b) or (c). This completes the proof of the theorem. \square

Note that in condition (a), namely (27a), setting $\gamma = 1/(1 + \delta_s)$ and $\hat{s} = s$ gives the same conditions on convergence for the proof of the IHT algorithm in [6]. Our result however generalises and shows that decreasing the step length can compensate for larger δ values. We choose to use the RIP rather than the modified-RIP, unlike the authors of [6], as it leads to easier implementation. To implement IHT one requires a measurement matrix scaled by $1/(1 + \delta_s)$, but it is unfeasible to perform this operation exactly. Hence by avoiding this scaling and choosing a deliberately smaller γ (which admittedly disguises some of the difficulty in this scenario) we can much more easily implement our algorithm.

Observe also that this theorem implies that as $\gamma \rightarrow 0$, the error due to the noise component in the model also goes to 0.

Strictly speaking, there is no real reason to require $\|\mathbf{r}^{(k+1)}\|_2 \leq \frac{1}{2} \|\mathbf{r}^{(k)}\|_2 + 2\gamma\sqrt{1 + \delta_{2s}} \|\mathbf{e}\|_2$, any value $0 \leq \alpha < 1$ with $\|\mathbf{r}^{(k+1)}\|_2 \leq \alpha \|\mathbf{r}^{(k)}\|_2 + 2\gamma\sqrt{1 + \delta_{2s}} \|\mathbf{e}\|_2$ would suffice, but perhaps offer significantly slower convergence. What happens to the convergence conditions if we allow a larger α ?

Assume we have $\|\mathbf{r}^{(k+1)}\|_2 \leq \alpha \|\mathbf{r}^{(k)}\|_2 + 2\gamma\sqrt{1+\delta_{2s}}\|\mathbf{e}\|_2$, then it follows that

$$\begin{aligned} \|\mathbf{r}^{(k)}\|_2 &\leq \alpha \|\mathbf{r}^{(k-1)}\|_2 + 2\gamma\sqrt{1+\delta_{2s}}\|\mathbf{e}\|_2 \\ &\leq \alpha^k \|\mathbf{r}^{(0)}\|_2 + 2\gamma\sqrt{1+\delta_{2s}}\|\mathbf{e}\|_2 \sum_{i=1}^k \alpha^i \\ &< \alpha^k \|\mathbf{r}^{(0)}\|_2 + \frac{2\gamma\sqrt{1+\delta_{2s}}}{1-\alpha} \|\mathbf{e}\|_2, \end{aligned}$$

which unfortunately threatens significant noise amplification, especially as α gets close to 1. Then the convergence criteria become

$$\begin{aligned} \text{(a) } \gamma &\geq \frac{1}{\delta_{s+2\hat{s}} - \delta_{s+\hat{s}} + 1}, \quad \text{and} \quad \gamma\delta_{s+2\hat{s}} \leq \frac{\alpha}{\sqrt{8}} < \frac{1}{\sqrt{8}} \approx 0.35, \quad \text{or} \\ \text{(b) } \gamma &< \frac{1}{\delta_{s+2\hat{s}} - \delta_{s+\hat{s}} + 1}, \quad \text{and} \quad \gamma(1 - \delta_{s+\hat{s}}) > 1 - \frac{\alpha}{\sqrt{8}} > 1 - \frac{1}{\sqrt{8}} \approx 0.65, \end{aligned}$$

which are slightly looser than before.

3.3 Convergence

If Algorithm 1 converges, it either converges to the correct sparse solution, or it converges to another sparse vector, but one that is not an (approximate – to some level of tolerance) solution to the equation $\mathbf{y} = \Phi\mathbf{x} + \mathbf{e}$. If the algorithm converges to an incorrect vector $\hat{\mathbf{x}}$, it is simple to test this since $\|\Phi\hat{\mathbf{x}} - \mathbf{y}\|_2 \gg \|\mathbf{e}\|_2$, and, if necessary, rerun the algorithm with different γ or \hat{s} estimates.

Suppose that the algorithm converges to $\hat{\mathbf{x}}$ and that we know \mathbf{y} exactly, i.e. we are investigating an equation of the form $\mathbf{y} = \Phi\mathbf{x}$. Note that the solution returned by the algorithm, $\hat{\mathbf{x}}$, is \hat{s} -sparse. Let $\mathfrak{P} \in \mathbb{R}^{n \times n}$ be a diagonal matrix that is the projection matrix onto the components specified by the support of $\hat{\mathbf{x}}$, i.e. onto the non-zero components of $\hat{\mathbf{x}}$. Observe that we have $\mathfrak{P}^2 = \mathfrak{P} = \mathfrak{P}^\top$ and $\mathfrak{P}\hat{\mathbf{x}} = \hat{\mathbf{x}}$. Then the solution $\hat{\mathbf{x}}$ obeys

$$\begin{aligned} \hat{\mathbf{x}} &= \mathfrak{P} [\hat{\mathbf{x}} + \gamma\Phi^\top\Phi(\mathbf{x} - \hat{\mathbf{x}})] \\ \Rightarrow \mathfrak{P}\Phi^\top\Phi\mathbf{x} &= \mathfrak{P}\Phi^\top\Phi\hat{\mathbf{x}} \\ &= \mathfrak{P}^\top\Phi^\top\Phi\mathfrak{P}\hat{\mathbf{x}}. \end{aligned}$$

Recall that the Moore-Penrose pseudo-inverse A^\dagger of a matrix A is given by, if it exists, $A^\dagger = (A^*A)^{-1}A^*$. Let $A = \Phi\mathfrak{P}$, then $\hat{\mathbf{x}}$ obeys

$$A^\top\Phi\mathbf{x} = A^\top A\hat{\mathbf{x}}.$$

If we can invert $A^\top A$, then $\hat{\mathbf{x}}$ is given by

$$\hat{\mathbf{x}} = (A^\top A)^{-1} A\Phi^\top\mathbf{x} = (\Phi\mathfrak{P})^\dagger \Phi\mathbf{x}.$$

This result is not at all surprising, it merely confirms that if we knew the support of \mathbf{x} , then we could find the solution by solving the least squares problem on this support.

4 Modifications

We propose several modifications to the MFR algorithm that increase both the success rate and the rate of convergence.

Algorithm 2 MFR with Polynomial Convergence

Input:

- The measurement matrix Φ .
- Observation vector \mathbf{y} .
- Estimate of sparsity \hat{s} of the vector \mathbf{x} .
- Step size γ .
- Tolerance parameter ε_{tol} .

Output:

- A vector $\hat{\mathbf{x}} \in \mathbb{R}^n$ that is \hat{s} -sparse.

```

1:  $\omega^{(0)} \leftarrow 1$ 
2:  $\mu \leftarrow \frac{\sigma_{\max}^2(\Phi) - \sigma_{\min}^2(\Phi)}{\sigma_{\max}^2(\Phi) + \sigma_{\min}^2(\Phi)}$ 
3:  $\mathbf{x}^{(0)} \leftarrow \mathbf{0}$ 
4:  $\mathbf{x}^{(1)} \leftarrow \Phi^\top \mathbf{y}$ 
5:  $k \leftarrow 1$ 
6: while  $\|\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}\|_2 \geq \varepsilon_{tol}$  do
7:    $\omega^{(k+1)} \leftarrow \frac{1}{1 - \omega^{(k)} \frac{\mu^2}{4}}$ 
8:    $\mathbf{x}^{(k+1)} \leftarrow \mathbb{H}_{\hat{s}} \left[ \mathbf{x}^{(k-1)} + \omega^{(k+1)} \left( \gamma \Phi^\top (\mathbf{y} - \Phi \mathbf{x}^{(k)}) + \mathbf{x}^{(k)} - \mathbf{x}^{(k-1)} \right) \right]$ 
9:    $k \leftarrow k + 1$ 
10: end while
11: return  $\mathbf{x}^{(k)}$ 

```

4.1 Accelerated Polynomial Convergence

We can easily apply the methods from Section 2.3 to increase the convergence rate of our algorithm to get Algorithm 2.

4.2 Least Squares

Another method to speed up convergence of the MFR algorithm, is to add a least squares minimisation step. The algorithm produces a sparse approximation to the solution, which we use as a method of selecting the columns of the solution that contain the non-zero data points. On this set of columns, say Γ , we then solve the least squares problem

$$\arg \min_{\mathbf{z}: \text{supp}(\mathbf{z})=\Gamma} \|\mathbf{y} - \Phi \mathbf{z}\|_2, \quad (39)$$

which has a convenient closed form solution. As this algorithm is deterministic, solving this problem twice when restricted to the same support, gives the same solution. Hence we will run the update step until the support of the largest \hat{s} components changes, and then solve the least squares problem restricted to this support. The full algorithm is given as Algorithm 3.

Furthermore, we can combine the polynomial variant of this algorithm with least squares.

Algorithm 3 MFR with Least Squares

Input:

- The measurement matrix Φ .
- Observation vector \mathbf{y} .
- Estimate of sparsity \hat{s} of the vector \mathbf{x} .
- Step size γ .

Output:

- A vector $\hat{\mathbf{x}} \in \mathbb{R}^n$ that is \hat{s} -sparse.

```

1:  $\mathbf{x}^{(0)} \leftarrow \mathbf{0}$ 
2:  $\Gamma^{(0)} \leftarrow \emptyset$ 
3:  $k \leftarrow 1$ 
4: while  $\|\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}\|_2 \geq \varepsilon_{tol}$  do
5:    $\hat{\mathbf{x}} \leftarrow \mathbf{x}^{(k)}$ 
6:    $\hat{\mathbf{x}} \leftarrow \hat{\mathbf{x}} + \gamma \Phi^\top (\mathbf{y} - \Phi \hat{\mathbf{x}})$ 
7:    $\hat{\mathbf{x}} \leftarrow \mathbb{H}_{\hat{s}}(\hat{\mathbf{x}})$  ▷ Prune
8:    $\Gamma \leftarrow \text{supp}(\hat{\mathbf{x}})$ 
9:   if  $\Gamma \neq \Gamma^{(0)}$  then
10:     $\mathbf{x}^{(k+1)} \leftarrow \Phi_\Gamma^\dagger \mathbf{y}$  ▷ Solve LS on  $\Gamma$ 
11:     $\Gamma^{(0)} \leftarrow \Gamma$ 
12:   else
13:     $\mathbf{x}^{(k+1)} \leftarrow \hat{\mathbf{x}}$ 
14:   end if
15:    $k \leftarrow k + 1$ 
16: end while
17: return  $\mathbf{x}^{(k)}$ 

```

4.3 Adaptive Step-length

So far we have only considered a step-length, γ , of fixed size, but inherently there is no reason why we cannot vary γ from one iteration to the next. Algorithms with a variable step-length are certainly well known, such as in soft-thresholding [16] and gradient pursuit methods [5].

What we propose is a greedy strategy, so that at each iteration the step length $\gamma^{(k)}$ is chosen so as to minimise a certain quantity. One obvious choice is to minimise the ℓ_2 norm of the residual, so that we would choose the step-length $\gamma^{(k)}$ to minimise $\|\mathbf{x} - \mathbf{x}^{(k)}\|_2$, i.e.

$$\gamma^{(k)} = \arg \min_{\gamma \geq 0} \left\| \mathbf{x} - \mathbb{H}_{\hat{s}} \left(\mathbf{x}^{(k-1)} + \gamma^{(k)} \Phi^\top (\mathbf{y} - \Phi \mathbf{x}^{(k)}) \right) \right\|_2.$$

Unfortunately, since we do not know \mathbf{x} , we cannot calculate this quantity. An alternative then is to minimise the residual in Φ -space, i.e. $\|\mathbf{y} - \Phi \mathbf{x}^{(k)}\|_2$, so at every iteration we choose $\gamma^{(k)}$ so as to minimise

$$\left\| \mathbf{y} - \Phi \mathbb{H}_{\hat{s}} \left(\mathbf{x}^{(k-1)} + \gamma^{(k)} \Phi^\top (\mathbf{y} - \Phi \mathbf{x}^{(k)}) \right) \right\|_2. \quad (40)$$

This gives us Algorithm 4.

Algorithm 4 MFR Algorithm with locally optimal γ

Input:

- The measurement matrix Φ .
- Observation vector \mathbf{y} .
- Estimate of sparsity \hat{s} of the vector \mathbf{x} .
- Tolerance parameter ε_{tol} .

Output:

- A vector $\hat{\mathbf{x}} \in \mathbb{R}^n$ that is \hat{s} -sparse.

```
1:  $\mathbf{x}^{(0)} \leftarrow \mathbf{0}$ 
2:  $k \leftarrow 1$ 
3: while  $\|\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}\|_2 \geq \varepsilon_{tol}$  do
4:    $\gamma^{(k)} \leftarrow \arg \min_{\gamma} \|\mathbf{y} - \Phi \mathbb{H}_{\hat{s}}(\mathbf{x}^{(k)} + \gamma \Phi^T(\mathbf{y} - \Phi \mathbf{x}^{(k)}))\|_2$ 
5:    $\mathbf{x}^{(k+1)} \leftarrow \mathbb{H}_{\hat{s}}(\mathbf{x}^{(k)} + \gamma \Phi^T(\mathbf{y} - \Phi \mathbf{x}^{(k)}))$ 
6:    $k \leftarrow k + 1$ 
7: end while
8: return  $\mathbf{x}^{(k)}$ 
```

This algorithm has two properties, firstly the ℓ_2 -norm of the residual is a non-decreasing function, as shown by the following lemma.

Lemma 11. *Let $\Phi \in \mathbb{R}^{m \times n}$ be a measurement matrix that obeys the RIP of order s and let $\mathbf{x} \in \mathbb{R}^n$ be the s -sparse signal we are trying to reconstruct given the measurements $\mathbf{y} = \Phi \mathbf{x}$. Using the MFR algorithm with adaptive step-length, Algorithm 4 where the step-length, is chosen at every iteration so as to minimise the residual in Φ -space, then the ℓ_2 -norm of the residual is a non-increasing function of the number of iterations.*

Proof. Clearly if $\gamma^{(k)} = 0$ then we have $\mathbf{x}^{(k)} = \mathbf{x}^{(k-1)}$ hence $\|\Phi \mathbf{r}^{(k)}\|_2 = \|\Phi \mathbf{r}^{(k-1)}\|_2$. Thus setting $\gamma^{(k)} = 0$ does not increase the norm of the residual, hence minimising this for $\gamma > 0$ can only further decrease $\|\Phi \mathbf{r}^{(k)}\|_2$. \square

Although Lemma 11 does not guarantee convergence to a vector $\hat{\mathbf{x}}$, it promises convergence to a vector $\Phi \hat{\mathbf{x}}$ that lies on a sphere around the measurements \mathbf{y} .

Secondly, provided that $\gamma^{(k)}$ meets the conditions of Theorem 2 then convergence is at least as fast, as the theorem does not depend on the particular value of γ used at every iteration.

5 Simulation Results

In this section we simulate a number of the algorithms we have discussed. All simulations use measurement matrices with Gaussian entries and the data signals were generated by randomly choosing a support, and then generating Gaussian entries with variance 1 for the non-zero components. All algorithms were terminated when the change in output from one iteration to the next was less than 10^{-7} using the ℓ_2 norm. We classed a simulation run as successful if it produced an output that had the correct support.

Table 1: Percentage of simulations that resulted in the correct vector being returned by the MFR algorithm with least squares for a matrix $\Phi \in \mathbb{R}^{50 \times 400}$. Numbers in bold give the maximum for that column. We see that we obtain a higher success rate if we underestimate the sparsity of the original signal.

ESTIMATED SPARSITY \hat{s}	TRUE SPARSITY			
	4	8	12	16
4	26%	-	-	-
8	94%	11%	-	-
12	93%	51%	5%	-
16	96%	55%	7%	0%
20	95%	53%	10%	2%
30	84%	26%	0%	0%
40	41%	6%	0%	0%
TOTAL	100%	79%	17%	2%

5.1 Success Rate

In Figures 1-2 we plot the success percentages when simulating some of the algorithms mentioned in this paper. In both cases we see that the MFR algorithm and its variants outperforms both ℓ_1 minimisation and the CoSaMP algorithm. The MFR algorithm with least squares (MFR + LS) is the best performing algorithm. However what this graph does not capture is the speed and number of iterations required for convergence, which can be seen in Figure 3.

In Table 1 we can see the advantage of choosing a sparsity estimate \hat{s} that is strictly larger than the true sparsity of the signal we are trying to reconstruct. In every case, the single best sparsity estimate is one that is larger than the real sparsity. This illustrates that it is in fact an advantage to overestimate the sparsity. This is also significant as if we underestimate the sparsity, the algorithm cannot succeed.

5.2 Convergence Rate

In Figure 3 we plot histograms of the number of iterations required before convergence for the three variants of our algorithms. We generated one thousand random Gaussian matrices Φ and one thousand sparse data vectors \mathbf{x} , so that each algorithm was fed the same input. We see clearly that the two modified versions converge significantly quicker than the plain MFR algorithm, and that the algorithm incorporating least squares converges faster still.

For a plot showing how the step-length affects the rate of convergence, see Figure 4.

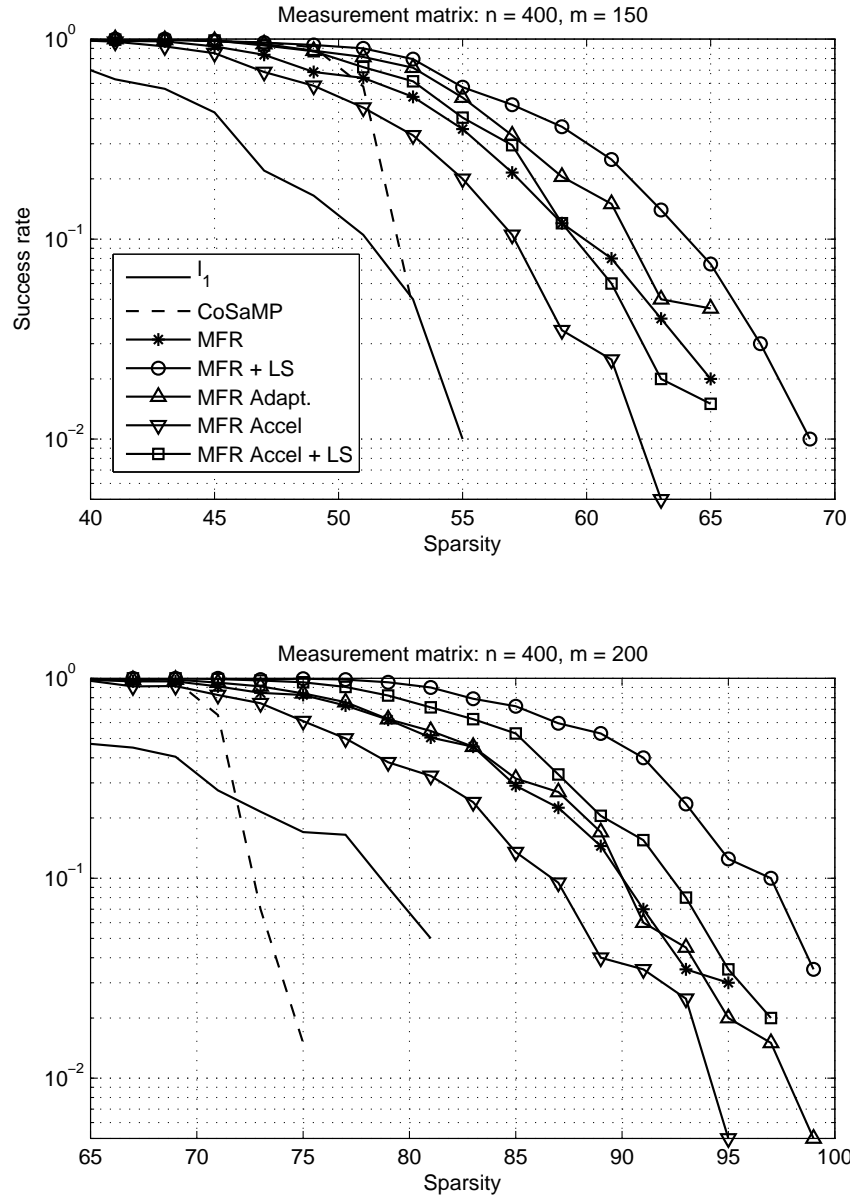


Figure 1: Success rates for simulating a number of algorithms with $n = 400$ and $m = 150$ (top) and $m = 200$ (bottom), using ℓ_1 minimisation, CoSaMP, MFR and its variants. In both cases MFR with Least Squares outperforms all other algorithms.

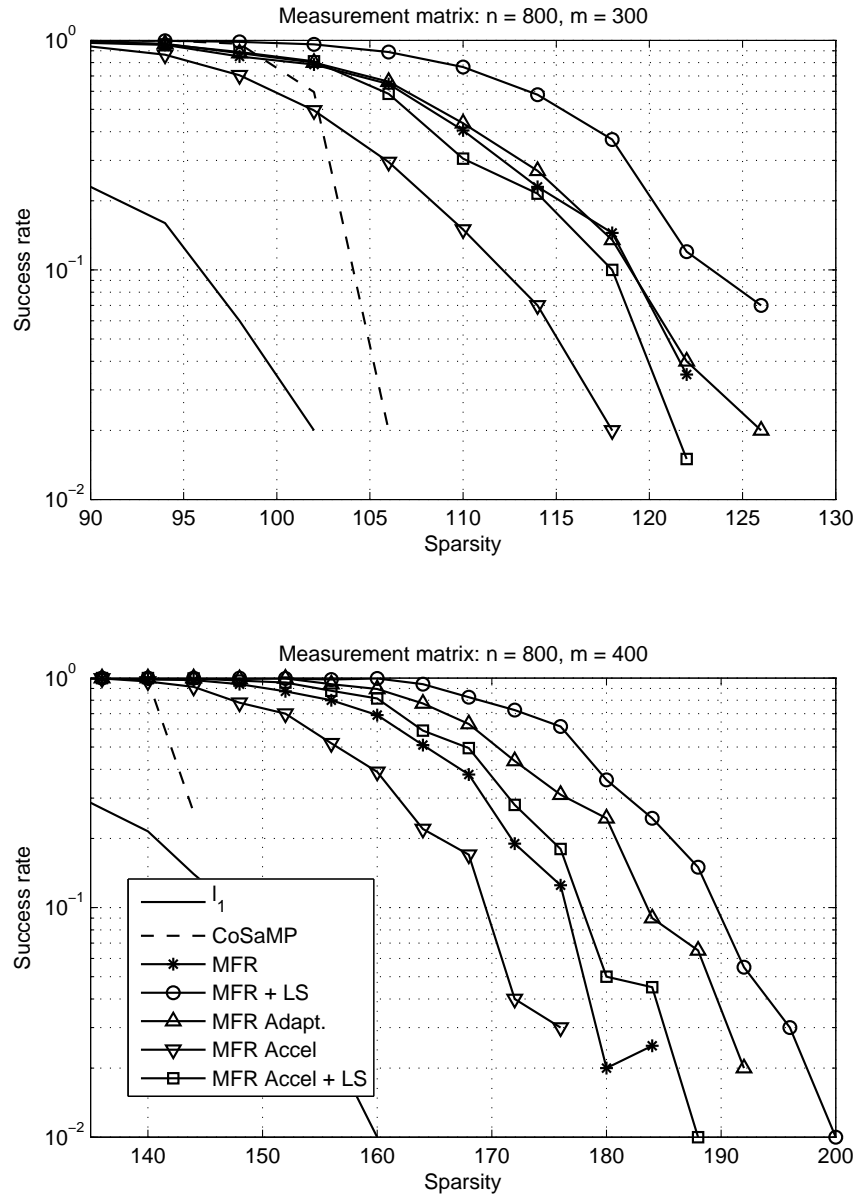


Figure 2: Success rates for simulating a number of algorithms with $n = 800$ and $m = 300$ (top) and $m = 400$ (bottom), using ℓ_1 minimisation, CoSaMP, MFR and its variants. In both cases MFR with Least Squares outperforms all other algorithms.

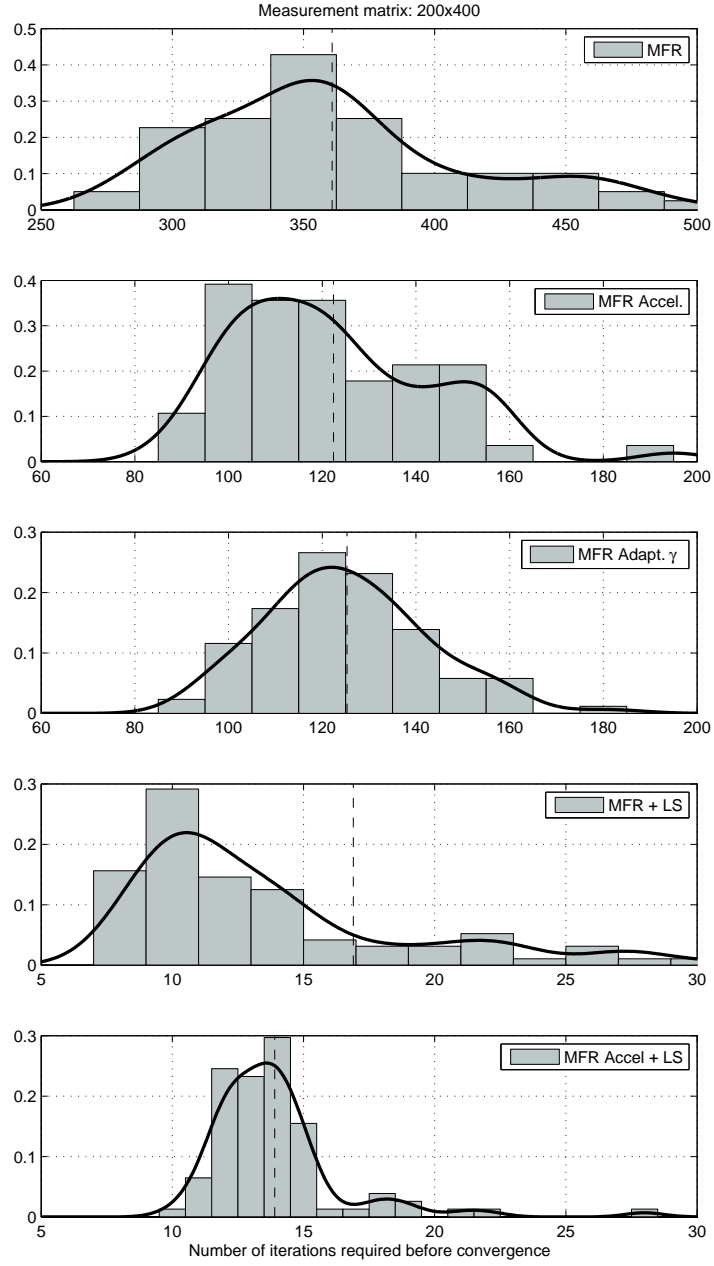


Figure 3: Histogram of the number of iterations required for convergence for the MFR algorithms. From top to bottom: MFR, MFR with polynomial acceleration, MFR with adaptive step-length, MFR with least squares and MFR with polynomial acceleration and least squares. The vertical dashed line shows the mean number of iterations required before convergence.

6 Comparison to Previous Algorithms

Here we present a detailed comparison of the MFR algorithm and its variants to several other reconstruction techniques.

The algorithm we have presented generalises the previous IHT algorithm. We need to be careful in directly comparing the two algorithms as they use slightly different measurement matrix structures. The IHT algorithm assumes a scaled matrix $\hat{\Phi}$, that is, let $\Phi \in \mathbb{R}^{m \times n}$ have RIP constant δ_s of order s , then the IHT algorithm reconstructs \mathbf{x} given the measurements

$$\mathbf{y} = \hat{\Phi}\mathbf{x} = \frac{\Phi}{1 + \delta_s}\mathbf{x},$$

whereas we operate directly with the original matrix Φ . This means that by setting $\gamma = \frac{1}{1 + \delta_s}$, dividing the measurements \mathbf{y} by $1 + \delta_s$ and putting $\hat{s} = s$ in our algorithm, the two algorithms are equivalent.

The advantage of allowing γ to be variable is phenomenal. We have already shown that by allowing γ to be smaller we can dramatically improve the reconstruction rate of the algorithm with low sparsity inputs. Alternatively with very sparse inputs we can increase the convergence rate by choosing the value of γ to be larger. Proposition 3 says that, under the hypothesis of the theorem, the error in the final signal is bounded by the term $4\gamma\sqrt{1 + \delta_{2\hat{s}}}\|\mathbf{e}\|_2$ where \mathbf{e} is the error in measuring the signal. Hence by taking a small value for γ we can decrease the effect of the error in the final output.

In comparison to the analysis of IHT, Theorem 2 offers a slightly better convergence result. Setting $\hat{s} = s$ and $\gamma = \frac{1}{1 + \delta_s}$ yields the identical condition to the main Theorem of [6], our analysis says that provided $\gamma \geq 1/(\delta_{s+2\hat{s}} - \delta_{s+\hat{s}} + 1)$, if $\delta_{s+2\hat{s}} > 1/\sqrt{32}$ then we can decrease γ and still get convergence. Alternatively, if $\delta_{s+2\hat{s}} < 1/\sqrt{32}$ then we can increase γ to get faster convergence, provided we could estimate the quantities $\delta_{s+\hat{s}}$ and $\delta_{s+2\hat{s}}$ accurately. The only way to do this currently, is to check all $\binom{n}{\tilde{s}}$ submatrices ($\tilde{s} = s + \hat{s}$ or $\tilde{s} = s + 2\hat{s}$ as appropriate) of Φ , which is computationally unfeasible and is in fact computationally equivalent to directly solving the original ℓ_0 problem, $\arg \min \|\hat{\mathbf{x}}\|_0$ subject to $\mathbf{y} = \Phi\hat{\mathbf{x}}$, directly. We see in Figure 4 how the rate of convergence increases dramatically with γ , which is what we predicted from Theorem 2.

By requiring a scaled measurement matrix, the implementation of the IHT algorithm becomes complicated. There is no known way to generate measurement matrices with particular RIP constants, although we can say with high probability they are bounded by a certain value, but to implement the IHT algorithm requires the matrix to be scaled by $1 + \delta_s$. Hence we must either estimate δ_s , which is not discussed in the original work, or calculate it explicitly, which means we might as well directly solve the ℓ_0 problem.

We have also proposed adaptively choosing $\gamma^{(k)}$ at every iteration. We have shown that doing so can dramatically decrease the error rate in reconstruction. It is ongoing work to see if there is a good way to estimate a better γ in terms of minimising the residual, without directly knowing the residual.

Another important difference to the IHT algorithm is that we discuss what happens if we threshold with a value that is strictly larger than the sparsity of the signal we are trying to reconstruct, that is, we keep more than s components of the signal non-zero. We often see that choosing a larger thresholding value dramatically improves the success rate at the expense of increasing the number of iterations required for convergence.

The algorithm we propose also appears on the surface to be very similar to the Gradient Pursuit algorithms discussed in [5]. Indeed, the directional update in gradient pursuit is the same as for both IHT and MFR, but the big difference is in how the sparsity constraint is enforced. For the gradient pursuit algorithms, a new dictionary element is added at every iteration, and

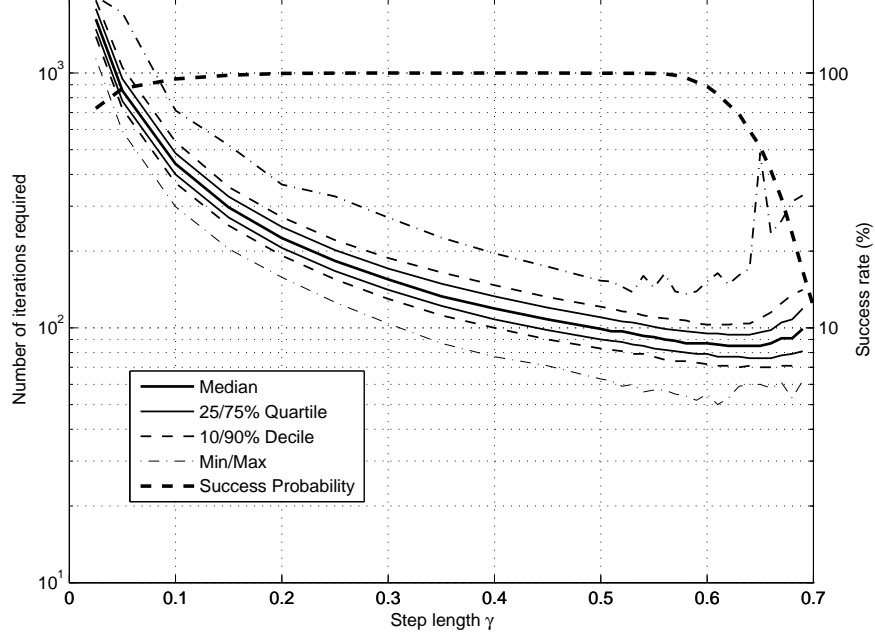


Figure 4: Number of iterations required before convergence using the MFR algorithm and different (fixed) step-lengths. This plot shows the quartiles for $\gamma \in \{0.05, 0.1 \dots, 0.7\}$. Observe that as γ increases the number of iterations required for convergence decreases, but the corresponding probability of success eventually falls.

once added, cannot be removed. In contrast, IHT and MFR make use of a pruning step, so at every iteration we keep only the most important (decided by the largest magnitude) dictionary elements, thus elements can be both added and removed.

More recent work in [9] uses the same paradigm as we do, i.e., selecting a possibly new support at each iteration, where the step-length is chosen to be the negative gradient of $\|\mathbf{y} - \Phi \mathbf{x}^{(k)}\|_2$, however their analysis is somewhat different to ours.

Incorporating a least squares step into the MFR algorithm makes it look similar to the CoSaMP algorithm in [27]. There are several significant differences though.

The CoSaMP algorithm solves the least squares problem over the vectors that have support size $3s$ where s is the sparsity of the true solution. The MFR algorithm typically considers a much smaller support size. The column or support selection for the CoSaMP algorithm is performed by merging the support of the the previous iterate and the “signal proxy”, that is $\Phi^T \Phi \mathbf{r}^{(k)} = \Phi^T \Phi (\mathbf{x} - \mathbf{x}^{(k)})$. In the case of the MFR algorithm, the frame reconstruction algorithm is used to select the support on which to solve the least squares problem.

Our work also addresses an area not explicitly discussed previously. We frame our reconstruction results in terms of the true sparsity of the signal and an estimate of the sparsity. Indeed we show via simulation that it can in fact be of benefit to under-estimate the sparsity of the signal we are trying to reconstruct.

7 Conclusion

In this paper we have demonstrated an iterative hard thresholding algorithm inspired by the frame reconstruction algorithm. In this work we have generalised the results of [6] and have shown how they can be used to guarantee convergence in a wider range of circumstances. Furthermore our algorithm significantly outperforms the IHT algorithm at larger s values (where s is the sparsity of the signal we are trying to reconstruct). At these points however, we operate beyond the range of values for which the theorem is applicable. That is, the δ and γ parameters do not satisfy either of our conditions because the sparsity level s of \mathbf{x} is too high. However we see from our simulations that even though the algorithms are not guaranteed to converge to the correct solution, they still often do.

We demonstrate several modifications to our algorithm; incorporating polynomial acceleration, adding a least squares step and using an adaptive algorithm to select the step-length. These modifications increase the performance, that is the rate of successful reconstruction and the rate of convergence.

For the least squares modification we show analytically that the algorithm will converge under the same requirements but possibly slightly slower, but empirically we observe that the algorithm in fact converges significantly faster. For the adaptive step-length modification we show that provided the step-length satisfies the requirements of Theorem 2 the algorithm will converge at least as fast.

References

- [1] Åke Björck. *Numerical Methods for Least Squares Problems*, volume 1. SIAM, 1996.
- [2] Richard G. Baraniuk, Volkan Cevher, Marco F. Duarte, and Chinmay Hegde. Model-based compressive sensing. *CoRR*, abs/0808.3572, 2008.
- [3] Richard G. Baraniuk and P. Steeghs. Compressive radar imaging. *Radar Conference, 2007 IEEE*, pages 128–133, April 2007.
- [4] Thomas Blumensath and Mike E. Davies. Gradient pursuits. *IEEE Trans. Signal Process.*, 56(6):2370–2382, June 2008.
- [5] Thomas Blumensath and Mike E. Davies. Iterative hard thresholding for compressed sensing. *CoRR*, abs/0805.0510, 2008.
- [6] Thomas Blumensath and Mike E. Davies. Iterative thresholding for sparse approximations. *Journal of Fourier Analysis and Applications, Special Issue on Sparsity*, 2008.
- [7] Thomas Blumensath and Mike E. Davies. Stagewise weak gradient pursuits part i: Fundamentals and numerical studies. *Preprint*, 2008.
- [8] Thomas Blumensath and Mike E. Davies. Stagewise weak gradient pursuits part ii: Theoretical properties. *Preprint*, 2008.
- [9] Thomas Blumensath and Mike E. Davies. Normalised iterative hard thresholding; guaranteed stability and performance. 2009.
- [10] Emmanuel J. Candès, Justin Romberg, and Terrence Tao. Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. *IEEE Trans. Inf. Theory*, 52(2):489–509, Feb. 2006.

- [11] Emmanuel J. Candès, Mark Rudelson, Terence Tao, and Roman Vershynin. Error correction via linear programming. In *FOCS*, pages 295–308, 2005.
- [12] Emmanuel J. Candès and Terence Tao. Decoding by linear programming. *IEEE Trans. Inf. Theory*, 51(12):4203–4215, Dec. 2005.
- [13] Emmanuel J. Candès and Terence Tao. Near-optimal signal recovery from random projections: Universal encoding strategies? *IEEE Trans. Inf. Theory*, 52(12):5406–5425, Dec. 2006.
- [14] Ingrid Daubechies. The wavelet transform, time-frequency localization and signal analysis. *IEEE Trans. Inf. Theory*, 36(5):961–1005, Sep 1990.
- [15] Ingrid Daubechies, M. Defrise, and C. De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on Pure and Applied Mathematics*, 57(11):1413–1457, 2004. 10.1002/cpa.20042.
- [16] Ingrid Daubechies, Massimo Fornasier, and Ignace Loris. Accelerated projected gradient method for linear inverse problems with sparsity constraint. *The Journal of Fourier Analysis and Applications*, 2007.
- [17] G. Davis. *Adaptive Nonlinear Approximations*. PhD thesis, New York University, 1994.
- [18] David Donoho, Iddo Drori, Victoria Stodden, Yaakov Tsaig, and Morteza Shahram. Sparse-Lab. Technical report, <http://sparselab.stanford.edu/>, 2007.
- [19] David L. Donoho. Compressed sensing. *IEEE Trans. Inf. Theory*, 52(4):1289–1306, April 2006.
- [20] David L. Donoho and J. Tanner. Counting faces of randomly projected polytopes when the projection radically lowers dimensions. Technical Report 2006-11, Department of Statistics, Stanford University, July 2006.
- [21] Gene H. Golub and Richard S. Varga. Chebyshev semi-iterative methods, successive over-relaxation iterative methods, and second order richardson iterative methods: Part i. *Numerische Mathematik*, 3(1):147–156, December 1961.
- [22] Gene H. Golub and Richard S. Varga. Chebyshev semi-iterative methods, successive over-relaxation iterative methods, and second order richardson iterative methods: Part ii. *Numerische Mathematik*, 3(1):147–156, December 1961.
- [23] Karlheinz Gröchenig. Acceleration of the frame algorithm. *IEEE Trans. Signal Process.*, 41(12):3331–3340, Dec 1993.
- [24] Christopher E. Heil and David F. Walnut. Continuous and discrete wavelet transforms. *SIAM Review*, 31(4):628–666, 1989.
- [25] Stéphane G. Mallat and Zhifeng Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Trans. Signal Process.*, 41(12):3397–3415, Dec 1993.
- [26] B. K. Natarajan. Sparse approximate solutions to linear systems. *SIAM Journal of Computing*, 24:227–234, April 1995.
- [27] D. Needell and Joel A. Tropp. Cosamp: Iterative signal recovery from incomplete and inaccurate samples. *Appl. Comp. Harmonic Anal.*, 26:301–321, 2008.

- [28] H. Nyquist. Certain topics in telegraph transmission theory. *Transactions of the A.I.E.E.*, pages 617–644, February 1928.
- [29] Claude E. Shannon. Communication in the presence of noise. *Proc. Institute of Radio Engineers*, 37(1):10–21, January 1949.
- [30] Joel A. Tropp and Anna C. Gilbert. Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Trans. Inf. Theory*, 53(12):4655–4666, Dec. 2007.
- [31] D. M. Young. *Iterative Solution of Large Linear Systems*. Academic Press, Inc., New York, 1971.